

REMARKS

I. Introduction

In response to the Office Action dated March 8, 2004, the claims have not been amended. Claims 1-36 remain in the application. Re-examination and re-consideration of the application is requested.

II. Prior Art Rejections

In paragraph (1) of the Office Action, claims 1-36 were rejected under 35 U.S.C. §103(a) as being unpatentable over Morrison, "Sams Teach Yourself MFC in 24 Hours", Copyright 1999, (Morrison).

Applicant acknowledges the indication of allowable claims, but respectfully traverses these rejections.

Specifically, independent claim 1 was rejected as follows:

In regard to Independent Claim 1 (and similarly independent Claims 11 and 21), Morrison teaches drawing bitmaps using the Microsoft Foundation Classes (MFC) class **CBitmap**. This class provides a member function **LoadBitmap ()** that is used to load a bitmap after a **CBitmap** object is created. To be loaded, the bitmap file (object data in this case) is inherently stored separate from the **CBitmap** object (instance). Compare with Claim 1 (and similarly with Claims 11 and 21), "... *storing the object data for the object separate from a file containing an instance of the object*". Morrison also teaches that one loads a bitmap by providing a bitmap resource identifier (i.e. pointer to a file name) to the **LoadBitmap ()** function (p. 162). Compare with Claim 1 (and similarly with Claims 11 and 21), "*obtaining a request to load the file*". The **LoadBitmap ()** function returns a nonzero value if it is successful and zero otherwise. Compare with Claim 1 (and similarly with Claims 11 and 21), "... *determining if the object data is available*". Morrison also teaches that after having loaded the bitmap data into the **CBitmap** object, one can paint the bitmap with the assistance of the member function of the CDC class called **BitBlt ()** (see steps involved on p. 162 to do this). Compare with Claim 1 (and similarly with Claims 11 and 21), "... *obtaining the object data and utilizing the object data to display a graphical representation of the object*".

Applicant traverses the above rejections. Specifically, Morrison fails to teach, disclose or suggest obtaining a request to load a file containing an instance of an object where object data for the object is stored separately;

Independent claims 1, 11, and 21 are generally directed to the use and storage of objects. Specifically, the claims provide that data for an object is stored separately from the location (i.e., the file) where an instance of the object is stored. The file containing the instance of the object is loaded. Upon loading, the invention determines if the object data (that is stored separately) is

available. If the object data is available, the object data is obtained (i.e., from the separate location). Further, the obtained object data is used to display a graphical representation of the object.

Morrison fails to teach, disclose, or suggest these various elements of Applicant's independent claims.

The rejection relies primarily on page 162 of Morrison that describes the LoadBitmap() function and CBitmap class. The rejection begins by referring to class CBitmap and a function LoadBitmap () of the CBitmap class. The rejection provides that the LoadBitmap() function is used to load a bitmap after a CBitmap object is created and provides that the bitmap file is inherently stored separate from the CBitmap object (instance). Accordingly, it is apparent that the Office Action is equating the CBitmap object to the claimed "instance of the object". In this regard, the Office Action is also equating the bitmap data that is loaded using the LoadBitmap () function with the object data (see Office Action).

With the above assertions in mind, the present claims provide for obtaining a request to load the file that contains the instance of the object. Accordingly, to teach the claim, Morrison must teach obtaining a request to load a file containing the CBitmap object (since the rejection provides that the claimed "instance" is equivalent to the CBitmap object). There is a clear distinction in the claims between loading the file containing the instance of the object and loading the data for the object. However, contrary to such a teaching, Morrison does not teach loading the CBitmap object whatsoever. Instead, Morrison merely describes loading a bitmap into the CBitmap object without loading the file containing the CBitmap object itself. The rejection appears confused as to which file is being loaded in accordance with the claims. Again, the claims specifically provide for obtaining a request to load the file containing the instance of the object (and not a request to load a file containing the object data for the instance). Without teaching these elements, Morrison cannot possibly teach the invention as claimed. In this regard, a specific function must be called to load the bitmap into the CBitmap class. Such actions are in opposition to loading a file containing the CBitmap class itself, followed by a determination if the object is available, and if available, obtaining the data and utilizing the data to display a graphical representation of the object (as claimed).

An additional claim element provides for determining if the object data is available. In rejecting this element, the rejection improperly states that the LoadBitmap () function returns a nonzero value if it is successful and zero otherwise. Applicant submits that there is no support for

such a statement in Morrison. Applicant conducted a search for the LoadBitmap function with a description of the function from Microsoft at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/bitmaps_4c34.asp (copy attached). As recited in the description, two return values are provided as part of the LoadBitmap () function:

Return Values

If the function succeeds, the return value is the handle to the specified bitmap.
If the function fails, the return value is NULL.

Thus, directly contrary to the statement in the Office Action, if the function fails, a NULL value is returned (and not a zero) and a handle is returned if successful (and not a nonzero value). Accordingly, the alleged support for the rejection does not exist.

In view of the above, Applicant submits that Morrison completely fails to teach, disclose, or suggest the invention as claimed.

The dependent claims provide significant limitations and details. In rejecting all of the dependent claims, the Office Action merely points out the advantages provided by the limitations in the claims. In this regard, Applicant appreciates the acknowledgment of the benefits provided by Applicant's invention. However, each dependent claim rejection then continues and merely provides that it would have been obvious to one of ordinary skill in the art. Applicant respectfully traverses such statements.

Firstly, the rejections of the dependent claims rely on impermissible hindsight. Under MPEP §2141.01, "The references must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention". Without any support based on the prior art, the assertion that it would have been obvious based on the benefit currently acknowledged today by the Examiner without any supporting description in Morrison (or other reference) is without merit.

Under MPEP 2142 and 2143, the burden is on the Examiner to establish a prima facie case of obviousness. As part of this prima facie case, three (3) criteria must be met. First, there must be some suggestion or motivation, either in the reference themselves, or in the knowledge generally available to one of ordinary skill in the art to modify the references. There is no such suggestion implicitly or explicitly in Morrison for any of the dependent claims. For example, dependent claim 2 provides for displaying an empty graphical representation if the object data is not available. Instead,

of displaying such an empty graphical representation, the function for loading the bitmap image would merely fail when the bitmap does not exist. There is no suggestion that any graphical representation would be displayed at all in this situation. Instead, the Office Action merely provides a benefit of allowing sequential loading of multiple objects in a graphics application to proceed without error. Applicant admits that this would be a great benefit. However, to allow the loading of multiple graphics objects to proceed without error, the programmer could provide for various error conditions (e.g., if the function fails - do nothing and continue, or if the function fails - display an error message to the user and continue, etc.) none of which would even remotely provide for displaying an empty graphical representation. Accordingly, to assume that a programmer of ordinary skill in the art would have thought of displaying an empty graphical representation is not plausible nor supported by the cited art.

The second requirement for establishing a *prima facie* case is that there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. In this regard, the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. *In re Vaack*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). The Office Action consistently fails to describe where the expectation of success and the teaching to make the claimed combination exists in the prior art.

In addition, under MPEP 2143.02, a statement that modifications of the prior art meet the claimed invention would have been " 'well within the ordinary skill of the art at the time claimed invention was made' " because the references relied upon teach that all aspects of the claimed invention were individually known in the art is not sufficient to establish a *prima facie* case of obviousness without some objective reason to combine the teachings of the references. *Ex parte Levengood*, 28 USPQ2d 1300 (Bd. Pat. App. & Inter. 1993). Such a statement is exactly what is asserted throughout the rejections of the dependent claims. Further, under MPEP 2144.03, "It is never appropriate to rely solely on "common knowledge" in the art without evidentiary support in the record, as the principal evidence upon which a rejection was based. *Zurko*, 258 F.3d at 1385, 59 USPQ2d at 1697". The rejection consistently fails to provide any evidentiary support in the record upon which the rejection is based.

In view of the above and consistent with the requirements of the MPEP, Applicant specifically requests evidentiary support for the various "obviousness" determinations asserted in the rejection of the dependent claims.

Moreover, the various elements of Applicant's claimed invention together provide operational advantages over Morrison. In addition, Applicant's invention solves problems not recognized by Morrison.

Thus, Applicant submits that independent claims 1, 11, and 21 are allowable over Morrison. Further, dependent claims 2-10, 12-20, and 22-36 are submitted to be allowable over Morrison in the same manner, because they are dependent on independent claims 1, 11, and 21, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-10, 12-20, and 22-36 recite additional novel elements not shown by Morrison.

III. Conclusion

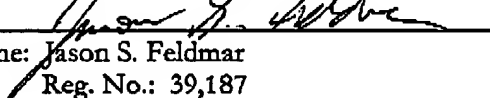
In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicant's undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: June 8, 2004

By: 
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/amb

G&C 30566.94-US-01